# Online algorithms with infused advice

Yuval Emek[1], Yuval Gil[1], **Maciej Pacut**[2], Stefan Schmid[2]

[1]Technion, Israel

[2]TU Berlin, Germany

# The success of randomized algorithms

"We do not assume anything about the distribution of the instances of the problem to be solved. Instead we incorporate randomization into the algorithm itself... It may seem at first surprising that employing randomization leads to efficient algorithm."

- Michael Rabin, 1976

Randomized algorithms often perform exceptionally well in practice

Why?

This talk: an advice model for randomized algorithms

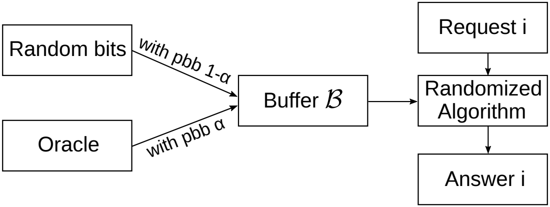
# Online algorithms and competitive analysis

- Stream of requests arrives one-by-one
- Algorithm answers without knowing the future
- Worst-case-oriented competitive analysis

$$CR(ALG) = \max_{\sigma} \frac{\mathbb{E}[ALG(\sigma)]}{OPT(\sigma)}$$

- The oblivious adversary (unaware of the algorithm's random choices)

# Our method: infused advice




(With probability $\alpha$ a clairvoyant oracle inserts "lucky" random bits)

# Infused advice: rationality

Randomized algorithms may enjoy good fortune (lucky random bits)

Infused advice measures impact of rounds with lucky random bits

# Our results

Analysis of three classic algorithms:

- **Caching**:
  RandomMark by Fiat, Karp, Luby, McGeoch, Sleator and Young
  (Journal of Algorithms 1991)
- **Uniform metrical task system**:
  algorithm by Borodin, Linial, Sacks
  (STOC 1987)
- **Online set cover**:
  primal-dual algorithm by Buchbinder and Naor
  (ESA 2005)

...and lower bounds for these problems

# Our results

| Problem | upper bound | lower bound |
|---|---|---|
| Caching | $\min\{2H_k, \frac{2}{\alpha}\}$ | $\min\{H_k, \frac{1}{\alpha}\}$ |
| Uniform MTS | $\min\{2H_n, \frac{2}{\alpha} + 2\}$ | $\min\{H_{n-1}, \frac{1}{\alpha}\}$ |
| Online set cover | $O(\min\{\log d \log n, \frac{\log n}{\alpha}\})$ | $\min\{\frac{1}{2} \log d, \frac{1}{2\alpha}\}$ |

# Our results

| Problem | upper bound | lower bound |
|---|---|---|
| Caching | $\min\{2H_k, \frac{2}{\alpha}\}$ | $\min\{H_k, \frac{1}{\alpha}\}$ |
| Uniform MTS | $\min\{2H_n, \frac{2}{\alpha}+2\}$ | $\min\{H_{n-1}, \frac{1}{\alpha}\}$ |
| Online set cover | $O(\min\{\log d \log n, \frac{\log n}{\alpha}\})$ | $\min\{\frac{1}{2}\log d, \frac{1}{2\alpha}\}$ |

Lower bounds assumptions:

- Lazy algorithms
- Not storing random bits from the current round for future rounds

(Without assumptions about advice size)

# Reminder of this talk

Why a new analysis method?

# Why randomized algorithms > deterministic algorithms?

Three reasons:

- Relaxed competitiveness: in expectation over random choices

# Why randomized algorithms > deterministic algorithms?

Three reasons:

- Relaxed competitiveness: in expectation over random choices
- Limited power of the adversary: hiding the current state

# Why randomized algorithms > deterministic algorithms?

Three reasons:

- Relaxed competitiveness: in expectation over random choices
- Limited power of the adversary: hiding the current state
- Increased power of the algorithm:
  opportunity to choose a cheaper execution path

# Why a new method?

"One of the primary goals of the mathematical analysis of algorithms is to provide guidance about which algorithm is the best for solving a given computational problem."

- Tim Roughgarden, 2020

# Why a new method?

"One of the primary goals of the mathematical analysis of algorithms is to provide guidance about which algorithm is the best for solving a given computational problem."

- Tim Roughgarden, 2020

- Online list access: Move-To-Front vs Frequency Count (2-competitive vs $\Omega(n)$-competitive)

# Why a new method?

"One of the primary goals of the mathematical analysis of algorithms is to provide guidance about which algorithm is the best for solving a given computational problem."

- Tim Roughgarden, 2020

- Online list access: Move-To-Front vs Frequency Count (2-competitive vs $\Omega(n)$-competitive)
- Caching: First-In-First-Out vs Least Recently Used (k-competitive vs k-competitive)

# Beyond worst-case analysis (online algorithms)

1. Distributional analysis
2. Random arrival order
3. Smoothed analysis
4. Diffused adversaries
5. Parameterized analysis (e.g. access graph, locality of reference)
6. Resource augmentation
7. Loose competitiveness
8. Advice and predictions

...and many more; see a book "Beyond Worst Case Analysis", Tim Roughgarden

# Why a new "beyond worst-case" analysis method?

Three categories of existing methods:

1. Change the input sequence (distributional analysis, random arrival)

# Why a new "beyond worst-case" analysis method?

Three categories of existing methods:

1. Change the input sequence (distributional analysis, random arrival)
2. Change the measure (resource augmentation, loose competitiveness)

# Why a new "beyond worst-case" analysis method?

Three categories of existing methods:

1. Change the input sequence (distributional analysis, random arrival)
2. Change the measure (resource augmentation, loose competitiveness)
3. Change the algorithm (advice, lookahead, predictions)

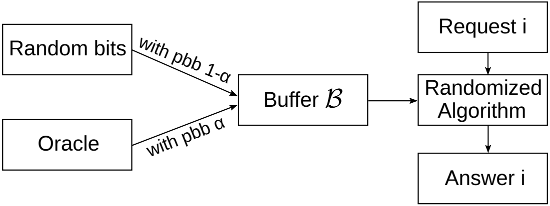
# Benefits of infused advice

- Can be applied to existing algorithms
- No assumptions about inputs
- Competitive ratio
- Lower bounds without bounding advice size

# Consequences

- Randomized algorithms design insights
- Beyond online algorithms: offline, distributed, streaming algorithms
- Augmenting existing randomized algorithms with predictions

# Summary




| Problem | upper bound | lower bound |
|---|---|---|
| Caching | $\min\{2H_k, \frac{2}{\alpha}\}$ | $\min\{H_k, \frac{1}{\alpha}\}$ |
| Uniform MTS | $\min\{2H_n, \frac{2}{\alpha} + 2\}$ | $\min\{H_{n-1}, \frac{1}{\alpha}\}$ |
| Online set cover | $O(\min\{\log d \log n, \frac{\log n}{\alpha}\})$ | $\min\{\frac{1}{2}\log d, \frac{1}{2\cdot\alpha}\}$ |